Example 29

[0364] The apparatus according to any of the examples 17 to 28 comprising one or more sensors (NAV**1**, G**1**) arranged to provide one or more words based on a detected environmental condition.

[0365] For the person skilled in the art, it will be clear that modifications and variations of the devices according to the present invention are perceivable. The figures are schematic. The particular embodiments described above with reference to the accompanying drawings are illustrative only and not meant to limit the scope of the invention, which is defined by the appended claims.

1-81. (canceled)

82. A method, comprising:

providing a query comprising one or more query words,

accessing a matrix containing co-occurrence data stored as vectors of the matrix,

determining a first auxiliary vector by identifying a vector of the matrix associated with a first query word,

forming a query vector by using the first auxiliary vector, and

determining a prediction by comparing the query vector with the vectors of the matrix.

83. The method according to claim **82**, further comprising:

determining a first difference between the query vector and a first vector of the matrix,

determining a second difference between the query vector and a second vector of the matrix, and

comparing the first difference with the second difference.

84. The method according to claim **81**, further comprising controlling a system based on the prediction.

85. The method according to claim **82**, further comprising controlling a user interface based on the prediction.

86. The method according to claim **82**, further comprising presenting a menu based on the prediction.

87. The method according to claim **82**, further comprising determining an auxiliary word from the first word by using a calendar, dictionary, electronic map, and/or tokenizing.

88. The method according to claim **87** wherein the auxiliary word is a semantic ancestor or a semantic descendant of the first word.

89. An apparatus comprising at least one processor, at least one memory including computer program code for one or more program units, the at least one memory and the computer program code configured to, with the processor, cause the apparatus to perform at least the following:

provide a query comprising one or more query words,

access a matrix containing co-occurrence data stored as vectors of the matrix,

determine a first auxiliary vector by identifying a vector of the matrix associated with a first query word,

form a query vector by using the first auxiliary vector, and

determine a prediction by comparing the query vector with the vectors of the matrix.

90. The apparatus according to claim **89** comprising computer program code configured to, with the processor, cause the apparatus to perform at least the following:

determine a first difference between the query vector and a first vector of the matrix,

determine a second difference between the query vector and a second vector of the matrix, and

compare the first difference with the second difference.

91. The apparatus according to claim **89** comprising computer program code configured to, with the processor, cause the apparatus to perform at least the following:

control the apparatus based on the prediction.

92. The apparatus according to claim **89** comprising computer program code configured to, with the processor, cause the apparatus to perform at least the following:

control a user interface based on the prediction.

93. The apparatus according to claim **89** comprising computer program code configured to, with the processor, cause the apparatus to perform at least the following:

present a menu based on the prediction.

94. The apparatus according to claim **89** comprising computer program code configured to, with the processor, cause the apparatus to perform at least the following:

determine an auxiliary query word from a first word by using a calendar, dictionary, electronic map, or tokenizing.

95. The apparatus according to claim **94** wherein the auxiliary word is a semantic ancestor or a semantic descendant of the first word.

96. A computer program product embodied on a non-transitory computer-readable medium, said medium including one or more computer-executable instructions that when executed by one or more processors cause a system to carry out at least the following:

to provide a query comprising one or more query words,

to access a matrix containing co-occurrence data stored as vectors of the matrix,

to determine a first auxiliary vector by identifying a vector of the matrix associated with a first query word,

to form a query vector by using the first auxiliary vector, and

to determine a prediction by comparing the query vector with the vectors of the matrix.

97. The computer program product of claim **96** wherein the instructions, when executed by one or more processors cause the system to carry out at least the following:

determine a first difference between the query vector and a first vector of the matrix,

determine a second difference between the query vector and a second vector of the matrix, and

compare the first difference with the second difference.

98. The computer program product of claim **96** wherein the instructions, when executed by one or more processors cause the system to carry out at least the following:

control the apparatus based on the prediction.

99. The computer program product of claim **96** wherein the instructions, when executed by one or more processors cause the system to carry out at least the following:

control a user interface based on the prediction.

100. The computer program product of claim **96** wherein the instructions, when executed by one or more processors cause the system to carry out at least the following:

present a menu based on the prediction.

101. The computer program product of claim **96** wherein the instructions, when executed by one or more processors cause the system to carry out at least the following:

determine an auxiliary query word from a first word by using a calendar, dictionary, electronic map, or tokenizing.